VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SIMULACE EIGRP PROTOKOLU V PROSTŘEDÍ OMNET++

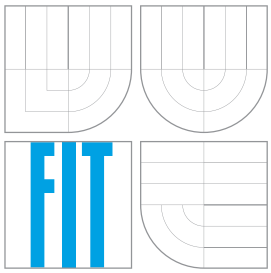
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN TLOLKA

BRNO 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SIMULACE EIGRP PROTOKOLU V PROSTŘEDÍ OMNET++

SIMULATION OF EIGRP PROTOCOL BEHAVIOR USING OMNET++

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARTIN TLOLKA
AUTHOR

VEDOUCÍ PRÁCE — Ing. ONDŘEJ RYŠAVÝ, Ph.D.
SUPERVISOR

BRNO 2009

Abstrakt

Práce se zabývá analýzou směrovacího protokolu EIGRP s výhledem na jeho začlenění do simulačního prostředí OMNeT++. Protokol EIGRP navržen firmou Cisco Systems je uzavřený standard, což představuje podstatný problém při implementaci jeho simulačního modelu. V této práci je uveden přehled chování protokolu podle dostupných materiálů a provedena analýza jednotlivých případů vycházejících z experimentů s reálnými zařízeními. Výsledkem je popis chování protokolu pro základní situace a identifikace vlastností, které by měla simulace splňovat.

Abstract

The present thesis deals with the analysis of EIGRP routing protocol for the purpose of integration of EIGRP simulation model in OMNeT++ environment. Protocol EIGRP defined by Cisco Systems is proprietary, which represents an obstacle in the implementation of a simulation model. In the present work, the description of a behavior of the protocol resembled from available information sources is given and then refined according the results obtained from experiments done with real network devices. The contribution of the work consists of a description of protocol behaviors in basics situations and the identification of properties that the simulation model should comply with.

Klíčová slova

simulace sítí, OMNeT++, INET, směrování, CISCO, EIGRP, DUAL

Keywords

network simulation, OMNeT++, INET, routing, CISCO, EIGRP, DUAL

Citace

Martin Tlolka: Simulace EIGRP protokolu v prostředí OMNeT++, bakalářská práce, Brno, FIT VUT v Brně, 2009

Simulace EIGRP protokolu v prostředí OMNeT++

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Ondřeje Ryšavého Ph.D.

.....
Martin Tlolka
17. května 2009

Poděkování

Děkuji vedoucímu mé práce panu Ing. Onřeji Ryšavému Ph.D. za odbornou pomoc, permanentní časovou dostupnost a v neposlední řadě za rady k vypracování této práce.

Obsah

Kapitola 1

Úvod

V rámci síťování je nezbytné zajistit, aby data dorazila v co možná nejkratším čase na správné místo určení. V případě komunikace mezi dvěma počítači je řešení jednoduché, ale v rámci vysokého počtu sítí a s tím souvisejícího přibývání počtu síťových rozhraní bylo nutné zavést systém, který zajistí, že data dorazí na místo určení. Toho se docílí tak, že směrovač (router) z paketu dat zjistí adresu příjemce, podívá se do směrovací tabulky a pošle paket na rozhraní, které je ve směrovací tabulce asociováno s adresou příjemce. V případě, že cílová adresa v tabulce není, posílá směrovač paket na bránu (implicitní cíl).

Jak ale směrovač směrovací tabulku získá? Existují dvě možnosti. Statické směrování vyžaduje ruční zadání adres, což je sice rychlé a levné, protože nevyžaduje žádné prostředky pro správu, ale pro větší sítě prakticky nepoužitelné. Každá změna se musí ručně upravit na všech směrovačích v síti. Druhou možnost představuje dynamické směrování, kde směrovacímu protokolu řekneme jen, o kterých sítích má šířit informace a protokol už zajistí, že změny v topologii se automaticky projeví na všech směrovačích, které daný protokol podporují a dynamické směrování mají nastaveno.

Mezi směrovací protokoly patří například RIP, RIPv2, EIGRP, OSPF.

Tato práce se má také zabývat počítačovými simulacemi sítí. Abychom vytvořili stabilní a udržovatelnou síť s potřebnými parametry, nemusíme kupovat drahá zařízení, zapojovat a testovat provoz či změny. K těmto experimentům slouží simulační nástroje. Pomocí nich lze jednoduše, levně a rychle nakonfigurovat situaci a sledovat požadované hodnoty. Nicméně je důležité ověřit si, jestli je dostupný nástroj vůči simulovaným událostem validní a měření odpovídá skutečnosti.

Cílem této práce je vytvořit modul pro simulační nástroj OMNeT++, který umožní simulovat směrování EIGRP a testovat tento způsob šíření informací o sítích.

V druhé kapitole se seznámíme s principem počítačových simulací, ukážeme si některé nástroje pro simulování sítí, a popíšeme si jejich výhody a nevýhody. Dále se zaměříme na prostředí OMNeT++ a jeho rozšíření INET. Lze zde nalézt podrobný popis instalace obou nástrojů. Také zde lze nalézt popis simulace, její spuštění a řízení. Ukážeme si také, jak vizualizovat výsledky.

Třetí kapitola se zabývá popisem protokolu EIGRP, získaným z uvedených zdrojů. Protože je EIGRP uzavřený protokol společnosti CISCO, a tudíž nemá zveřejněnou podrobnou specifikaci, bylo během pokusu o implementaci zjištěno, že je tento popis nedostatečný, a po několika iteracích hledání a přepisování kódu jsem se rozhodl, že je třeba provést analýzu chování přímo na reálných přístrojích CISCO.

Další kapitola se zabývá analýzou protokolu EIGRP metodou reverzního inženýrství. Je zde popsáno několik situací, kde je ze zjištěné komunikace a změnách na směrovačích zjišťo-

váno pravděpodobné chování tohoto protokolu. Celé chování je pak shrnuto do diagramů, podle kterých by mohl být protokol implementován. Dále se kapitola zabývá nutnými změnami v prostředí INET, které by umožňily vytvoření a začlenění modulu EIGRP.

V závěru lze nalézt zhodnocení získaných poznatků a další možné pokračování.

Kapitola 2

Simulace sítí

Kapitola se zabývá simulacemi v oblasti sítí. Je zde uvedeno několik simulačních nástrojů a jejich porovnání, přičemž se klade důraz na OMNeT++ s rozšířením INET Framework. Kapitola dále popisuje práci s těmito nástroji.

2.1 Diskrétní simulace

Cílem simulace je analýza chování systému v závislosti na vstupních veličinách a hodnotách parametrů. Pro provádění simulace je třeba si vytvořit simulační model, na kterém lze experimenty provádět. Ten je programovou implementací abstraktního modelu, který představuje zjednodušený reálný model, jehož chování chceme pozorovat. Pro vytvoření abstraktního modelu je nutné určit, které prvky systému jsou pro experimenty důležité a které můžeme zanedbat. Pokud máme abstraktní model, obvykle už není problém převést jej na simulační a začít s modelem experimentovat.

Diskrétní simulace je taková simulace, ve které změny v systému probíhají skokově. Mohou mít spojitý i diskrétní čas. V rámci číslicových počítačů se z důvodu ne nekonečné přesnosti spojitý čas diskretizuje s vhodně malým krokem změny času.

2.2 Přehled některých simulačních nástrojů

2.2.1 NS-2

Network simulation version 2 (NS-2) [3] je simulační nástroj pro diskrétní simulace v rámci počítačových sítí. Projekt začal jako varianta REAL network simulator v roce 1989. Je naprogramován v jazyce C++, který také slouží pro operace s daty v simulaci. Vlastní řízení simulace ja pak popsáno objektovým jazykem OTcl a uloženo v souborech .tcl. Mezi jinými umožňuje například budování modelu sítě, generování provozu, chyb. Podporuje drátovou i bezdrátovou technologii spojení. Jednotlivé prvky sítě reprezentují tzv. uzly (Nodes), které jsou propojeny linkami s nastavitelnými vlastnostmi (delay, bandwith, fronta). Více o tomto simulátoru najdete na stránkách projektu.

Výhody a nevýhody

Výhodou je možnost psaní vlastních modulů. Nevýhodou NS-2 je absnence ip knihovny, směrování, adresování. Dále pak slabá podpora multicastu, nedostatatečná schopnost vizualizace a celková složitost.

2.2.2 Packet Tracer

Packet tracer je výukový program společnosti CISCO. Obsahuje propracované grafické ovládání, které se ztotožňuje se symbolikou výukových materiálů CISCO akademie. Prvky sítě vytvoříte jednoduchým přetažením jejich vzorů do simulačního pole, kde je možné je nakonfigurovat. Prvky simulují i konzoli, která se chová podobně, jako na skutečných rozhraních. Pro začátečníky v oblasti sítí je to podle mého názoru jeden z nejlepších nástrojů.

Výhody a nevýhody

Výhodou je jednoduchá instalace a intuitivní ovládání, simulace konzolového ovládání, grafické prostředí. Hlavní nevýhodou je to, že se jedná o uzavřený systém společnosti CISCO, nelze vytvářet nové moduly a není volně šiřitelný.

2.2.3 OMNeT++

OMNeT++ je silný nástroj pro simulování počítačových systémů. Mezi jeho přednosti patří vyspělé grafické prostředí, flexibilita a vestavěná simulace v jádře, která proces simulace významně urychluje. Jeho primární použití je v počítačových komunikacích a sítích. Protože je ale velice flexibilní, umožňuje simulaci i v takových odvětvích jako jsou například výrobní procesy, hardwarová architektura atd.

Simulace je popsána jazykem NED. Nastavení simulace (například čas) je v souboru omnetpp.ini Prvky simulace (moduly) jsou napsány v C++.

Výhody a nevýhody

Výhodou je rozšíření INET Framework, které nabízí množství síťových modulů a jednoduchost tvorby modulů vlastních. Je volně šiřitelný. Nevýhodou je horší řízení simulace za běhu.

2.3 OMNeT++ a INET Framework

2.3.1 Rozšíření INET

INET framework implementuje do prostředí OMNeT++ protokoly IPv4, IPv6, TCP, UDP a několik aplikačních modelů.

2.3.2 Instalace MS Windows XP

OMNeT++

1. Nainstalovat Microsoft Visual Studio 2005
2. Spustit skript pro nastavení lokálních proměnných Visual studia v „cesta k visual studiu“/VC/vcvarsall.bat
3. Ze stránek OMNeTU [4] stáhnout a spustit binární instalaci OMNeT++ v 3.3
4. Rozbalit archiv
5. Nastavit proměnné prostředí Windows: do proměnné PATH přidat cestu „cesta k adresari omnet++“/bin

6. Otestovat pomocí rundemo.bat v adresáři sample (vyzkoušet jednotlivé simulace)

INET Framework

1. Ze stránek OMNeTU [4] v sekci simulačních modelů stáhnout INET Framework v20061020
2. Rozbalit archiv
3. Příkazem makemake vytvořit Makefile
4. Pomocí příkazu nmake zkompilovat
5. Otestovat pomocí skriptu rundemo.bat v adresáři Examples (vyzkoušet jednotlivé simulace)

2.3.3 Instalace Linux

Postup předpokládá gcc překladač v4.3 a program make.

OMNeT++

1. Ze stránek OMNeTu [4] stáhnout a spustit binární instalaci OMNeT++ v 3.3p1 source
2. Rozbalit archiv
3. Nainstalovat pomocné programy: sudo apt-get install bison flex blt lmodern giftrans doxygen libxml2-dev graphviz imagemagick tcl8.4 tk8.4 tcl8.4-dev tk8.4-dev
4. Spustit ./configure v adresáři s omnetem
5. Podle doporučení vložte exporty z výpisu do souboru .profile v HOME adresáři
6. Spustit make
7. Otestovat pomocí rundemo v adresáři samples

INET Framework

1. Ze stránek OMNeTu [4] v sekci simulačních modelů stáhnout INET Framework v20061020
2. Rozbalit archiv
3. Příkazem makemake vytvořit Makefile
4. Přepsat v souboru inetconfig cesku k nastaveni opmnetppconfig
5. Pomocí příkazu make zkompilovat
6. Přidat inet/bin do proměnné LD_LIBRARY_PATH
7. Otestovat pomocí skriptu rundemo.bat v adresáři Examples (vyzkoušet jednotlivé simulace)

2.3.4 Popis simulace

Topologie simulace je popsána jazykem NED. Ten popisuje jednotlivé moduly a submoduly, které se do sebe mohou zanořovat a vytvářet tak transparentní simulační model. Základním stavebním kamenem je modul `simple`, který popisuje jeden základní stavební typ. Lze mu určit jaké bude mít vstupní a výstupní porty (`gates`) a parametry (`parameters`). Vstupní porty mohou být zadávány i jako vektory s pevně danou, či volnou délkou. Vyšším stavebním prvkem jsou moduly (`modules`), kterým lze také přiřadit parametry. Moduly mohou obsahovat submoduly (`submodules`) a jejich důležitou vlastností je, že obsahují informace o propojení mezi submoduly (`connections`). V rámci propojení prvků lze definovat kanály (`chanels`) a nastavovat tak jednotlivým propojením důležité parametry, jako je šířka pásma, délka zpoždění atd. Pro jednoduché simulace lze také parametry zadávat přímo k připojení. Jazyk NED také umožňuje zápis podmínek a cyklů, má i své vestavěné funkce např. `sizeof()` a hojně používané stochastické funkce vhodné pro generování chyb, příchodů atp. Dalším prvkem simulace jsou zprávy definované v souborech .msg. Takto lze definovat zprávy, které si budou prvky mezi sebou zasílat. Také zprávy mohou do sebe zapouzdřovat jiné objeky. OMNeT++ z těchto souborů vytvoří třídy v C++. Další důležitou vlastností je možnost vkládat již definované moduly pomocí direktivy `import`.

Jak jsem již uvedl dříve, chování modulů je popsáno jazykem C++, kde lze reagovat na vzniklé události jako je např. příchod zprávy, výpisy do logů, načítání nastavení a další. Program OMNeT++ tyto soubory sám poskládá do jednoho celku.

Když máme nadefinované a poskládané jednotlivé moduly je třeba ještě nadefinovat síť, čili vlastní instanci modulu. Síti lze také nastavovat parametry. Definice síťe je nutná pro spuštění simulace.

Dále je pro spuštění simulace nezbytné nadefinovat parametry simulace v souboru omnetpp.ini. Tento soubor pak lze editovat bez nutnosti opětovného překladu.

2.3.5 Řízení simulace

V soubory scenario.xml je popis změn v čase. Řídíci modul pošle v nastaveném čase zprávu o změně s příslušnými parametry.

2.3.6 Překlad a spuštění simulace

Před samotným spuštěním simulace je nutné pomocí příkazu `opp_makemake`/`opp_nmakemake` vygenerovat třídy jednotlivých modulů a soubor Makefile, nutný pro kompilaci. Pomocí příkazu make/nmake simulaci přeložíme do spustitelného programu. Po úspěšném přeložení lze simulaci spustit.

2.3.7 Vizualizace výsledků

Běh simulace je animován (zasílání zpráv). U jednotlivých modulů lze zobrazit detaily. Takto lze jednoduše sledovat celou hiearchii modulů. Pomocí funkce `WATCH()` lze sledovat hodnoty parametrů modulů v jejich detailu.

Pomocí tříd `cLongHistogram` a `cOutVector` lze získávat statistiku simulace. Ta se ukládá do souborů .sca, ze kterých není problém data exportovat, nebo pomocí OMNeT++ nástrojů scalars a plove zobrazit.

Kapitola 3

EIGRP směrování

Tato kapitola popisuje problematiku směrování v sítích, zvláště pak směrovací protokol EIGRP. Informace zde uvedené jsou zpracovávány z veřejně dostupných materiálů. Jsou zde uvedeny vlastnosti i chování. Dále jsou zde také popsány důležité tabulky, výpočet metriky a DUAL algoritmus pro hledání ztracené cesty.

3.1 Vlastnosti EIGRP

Enhanced Interior Gateway Routing Protocol [2]. Jedná se o patentovaný CISCO protokol, který vychází ze staršího IGRP protokolu. Na rozdíl od něj však podporuje beztřídní směrování. Vyznačuje se rychlým šířením změn a snadnou konfigurací. Redukuje spotřebu pásma. Je nezávislý na ostatních směrovacích protokolech, ale umožňuje i meziprotokolové směrování. Předchází tvoření smyček. Využívá technik jako je split horizon, poison reverse a Hold Down timer. Pro hledání cest využívá DUAL algoritmus. Podporuje autosumarizaci.

Sítě jsou rozděleny do autonomních systémů, což jsou oblasti, ve kterých si směrovače mezi sebou vyměňují informace. Autonomní systém je reprezentován číslem (AS číslo) a toto číslo je pak obsaženo v nastavení směrovacího protokolu na směrovači a v EIGRP paketu.

3.2 Chování EIGRP

Pro komunikaci EIGRP využívá protokol RTP [2], který zajišťuje spolehlivý multicast na adrese 224.0.0.10 a portu 88. Pro uchování informací o přímo připojených směrovačích (sousedech) používá tabulku sousedů. Pro uchování všech možných cest využívá topologickou tabulku. Nejlepší cety ukládá do směrovací tabulky.

3.2.1 Výpočet metriky

Výpočet metriky cesty určuje, která cesta bude vybrána jako nejlepší (successor) a která jako záložní (feasible successor). Platí, že čím nižší číslo, tím lepší cesta. Rovnice pro výpočet metriky je následující:

$$256 * ([K1 * Bw + K2 * Bw/(256 - Load) + K3 * Delay] * [K5/(Reliability + K4)]) \quad (3.1)$$

kde K1 - K5 jsou nastavitelné parametry (obvykle 10100). Pokud je K5 = 0, je poslední zlomek ignorován:

$$256 * (K1 * Bw + K3 * Delay) \quad (3.2)$$

Dále pak Bw je šířka pásma, Load je vytíženost linky, Delay je zpoždění na lince, Reliability je spolehlivost linky. Parametry Ki lze upravovat váhu jednotlivých vlastností linky.

3.2.2 EIGRP paket

EIGRP paket obsahuje mimo obvyklých (kontrolní součet, délka) následující informace:

- Verzi protokolu
- Operační kód:
  - 1 UPDATE: akualizace topologie
  - 3 QUERY: dotaz (existuje jiná cesta?)
  - 4 REPLY: odpověď
  - 5 HELLO: pro hledání sousedů
  - 6 IPX SAP
- ACK: 32bit sekvence posledního paketu přijatého od souseda (HELLO paket s nenulovým ACK je ACK)
- AS číslo autonomního systému
- TLV podle toho, jestli jde o interní směrování, nebo externí
- parametry K1 - K5
- Next Hop: adresa dalšího skoku
- Delay (součet)
- Bandwith (nejmenší)
- MTU (nejmenší)
- Hop count: počet skoků do cíle
- Spolehlivost
- Vytíženost
- Cílová síť

Pro externí směrování ještě

- Původní AS
- Tag: pro mapování cest
- Metriku externího protokolu
- ID externího protokolu
- Příznaky

3.2.3 Zjišťování sousedů

Pro zjišťování sousedů EIGRP využívá HELLO pakety (s operačním kódem 5). Pokud není nutné použít unicast adresování, využívá EIGRP pro zjišování sousedů multicast na adrese 224.0.0.10. Na sítích typu LAN a na rychlých WAN posílá HELLO paket s intervalem 5s, na pomalé WAN s intervalem 60s. Pokud soused pošle ACK nebo vrátí HELLO paket s nenulovým ACK v případě multicastových sítí, pošle mu směrovač UPDATE pakety s informacemi o topologii (všechny cesty). HELLO pakety obsahují Hold Timer, který se spustí před označením souseda za nedostupného, a označí jej až když je roven trojnásobku HELLO intervalu.

Tabulka sousedů obsahuje:

- Pořadí, v jakém byli sousedé objeveni
- IP souseda
- Interface, na kterém byl soused objeven
- Hold Time v sekundách
- Up Time v sekundách
- SRTT: průměrný čas v ms mezi odesláním paketu a přijetím potvrzení
- RTO: čas v ms, který čeká směrovač na potvrzení
- počet paketů QUEUE ve frontě
- Sekvenční číslo posledního obdrženého paketu

Aby šetřil síťovou komunikaci, využívá EIGRP techniku split horizon, která zajistí zajistí, že směrovač neposílá informace o síti na rozhraní, které pro dosažení sítě používá. Dále pak využívá techniku poison reverse. Poison reverse posílá na rozhraní, které vede do cíle cesty, UPDATE paket s informací, že tato síť je přes něj nedostupná. Vyznačuje se nekonečnou hodnotou u parametru Delay.

3.2.4 Topologie sítě

Pro každou cestu, kterou se směrovač dozví pakety UPDATE, vypočítá vlastní vzdálenost tak, že k hlášené ceně (reported distance), která je v paketu, připočítá cenu cesty ke směrovači, který paket poslal. Cesta s nejmenší cenou (metrikou) je označena za successor a je uložena ve směrovací tabulce. Ostatní cesty s vyšší metrikou jsou označeny jako feasible successor a uloženy v topologické tabulce pro případ, že by se successor selhal. Aby se předešlo smyčkám ve směrování, jsou do topologické tabulky ukládány jen ty cesty, kdy reported distance je menší než feasible distance.

Topologická tabulka obsahuje následující informace:

- Feasible distance: nejnižší metrika
- Feasible successors: záložní cesty do cíle
- Jejich metriky
- Lokální metriky cesty přes každý feasible successor

- rozhraní (interface), na kterém je feasible successor připojen

Feasible successor s nejnižší lokální metrikou je označen za successor a uložen do směrovací tabulky.

Pokud se linka k successoru přeruší (vyprší Hold-down timer), najde se další cesta s nejmenší metrikou, což nevyžaduje síťový provoz. Pokud v tabulce k danému cíly už žádná záložní cesta není, pošle směrovač QUERY paket všem sousedům a ptá se na alternativní cestu. Když se směrovací informace změní, pošle UPDATE paket jen těm sousedům, kterých se změna týká.

3.2.5 DUAL algoritmus

Diffusing UPDATE Algorithm. Tento algoritmus se vyznačuje tím, že sdílí výpočet s ostatními směrovači. Směrovač jen přidá vzdálenost přímo připojených sousedů. Taktéž změny v topologii posílá jen těm sousedům, kterých se změna týká. Tímto šetří pásmo i výpočetní čas procesoru směrovače.

Principy DUAL:

- Ztráta detekce souseda se děje v konečném čase
- Zprávy jsou přijaty správně a ve správném pořadí doručovány v konečném čase
- Zprávy jsou zpracovávány ve správném pořadí a konečném čase

Pasivní stav

V pasivním stavu je směrovač tehdy, když más successory pro každou známou síť v tabulce topologií. Topologická tabulka se mění, jen když se smění cena linky, stav, nebo po obdržení UPDATE, QUERY a REPLY paketů. Pokud se změní successor, pošle se změna pomocí UPDATE paketu, ale směrovač zůstává v pasivním stavu.

Aktivní stav

Pokud neexistuje žádný successor do některé ze sítí, přejde směrovač do aktivního stavu a pošle QUERY pakety všem sousedům, aby od nich získal směrovací informace. Po přijetí QUERY paketu soused zjišťuje, jestli je odesílatel successor pro dotazovanou síť. Pokud není, pošle mu svůj successor, nebo odpoví, že síť je nedostupná. Pokud odesílatel je successor, přejde také do aktivního stavu a začne hledat jinou cestu, kterou pak pošle odesílateli. Pokud žádnou nenajde, odpoví, že síť je nedostupná.

Směrovač se nachází v aktivní stavu, dokud neobdrží odpovědi ode všech sousedů. Pro velké sítě to může být problém, protože značně prodlužuje čekání na výsledek. Pokud DUAL běží déle než 3 minuty, je směrovač označen za SIA (stuck in active), soused je vyjmut z tabulky sousedů a metrika pro jeho cestu je nastavena na nekonečno. Pokud odpověď přijde po vypršení limitu, je soused opět vrácen do tabulky sousedů.

Pravidla zpracování požadavků [1]

Když směrovač obdrží QUERY paket od souseda, aplikují se následující pravidla:

Dotaz od	Stav směrovače	Akce
soused, který není successor	pasivní	Odpoví aktuálním successorem.
soused successor	pasivní	Snaží se vyhledat nový successor a vrátí jej. Pokud neuspěje, označí cíl za nedosažitelný a informuje zbylé sousedy.
soused, který není successor	aktivní	Pokud zná successor k cíli, pošle informace o cestě, jinak označí cíl nedosažitelný a odpoví.
soused successor	aktivní	Snaží se vyhledat nový successor a vrátí jej. Pokud neuspěje, označí cíl za nedosažitelný a informuje zbylé sousedy.
jakýkoliv soused	přes souseda není cesta	Odpoví nejlepší cestou.
jakýkoliv soused	před dotazem neznámý	Odpoví, že cíl je nedosažitelný.

Kapitola 4

EIGRP v OMNet++

Kapitola popisuje analýzu protokolu a jeho možné začlenění do prostředí OMNet++

4.1 Analýza

Protože je EIGRP uzavřený protokol, bylo nutné provést podrobnou analýzu. Analýza je prováděna metodou reverzního inženýrství a to tak, že budeme sledovat probíhající komunikaci mezi CISCO směrovači. K tomu bylo využito monitorování portů na přepínači, kde jsou připojeny rozhraní směrovačů, na kterých je komunikace sledována a počítač, který sledování provádí pomocí programu Wireshark. Topologie zapojení popisuje obrázek 4.1

Dále popsané scénáře jsou prováděny a měřeny na této topologii. Na počátku mají směrovače tovární nastavení a změny nastavení jsou popsány v rámci jednotlivých scénářů.

Analýza je prováděna na cisco směrováčích CISCO 2811 s IOS verzí 12.4.

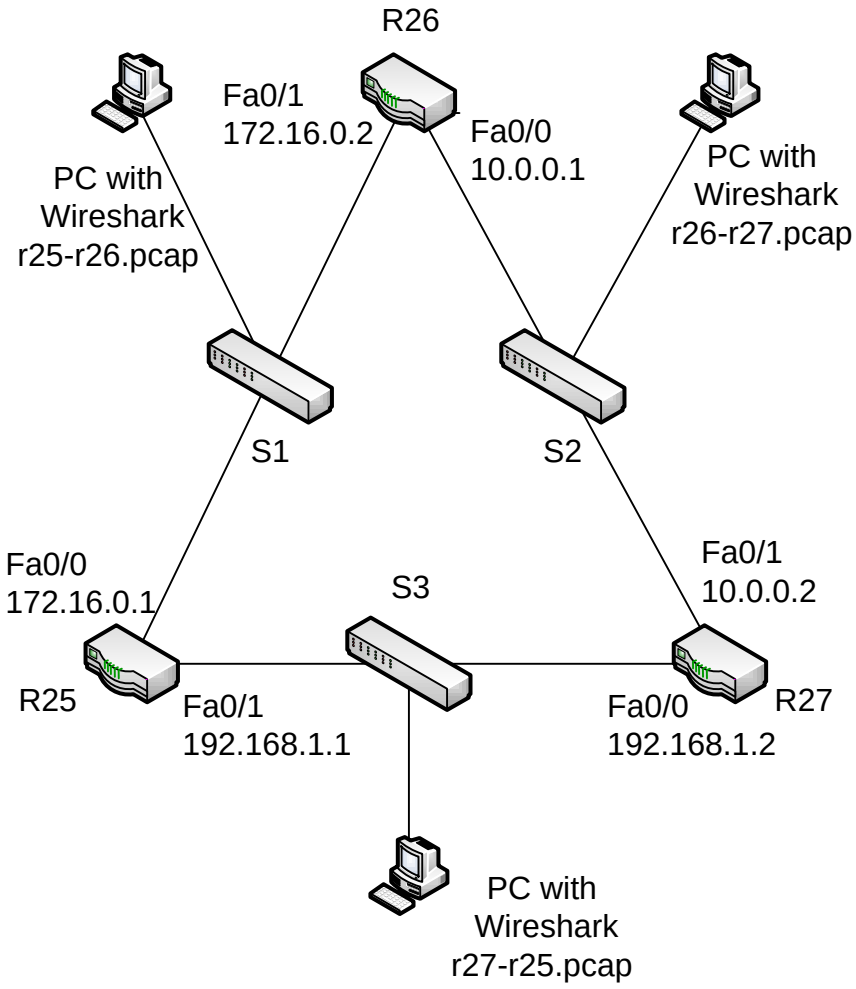
Na všech směrovačích jsou zapnuty ladící výpisy a jejich záznam je uložen na přiloženém CD. Na tomto CD je taktéž přiložen záznam příchozích a odchozích paketů zachycených programem Wireshark. Pojmenování souborů je x-y.pcap, kde x je směrovač na jehož rozhraní naslouchámе a y je sousední směrovač.

4.1.1 Zjišťování sousedů

Scénář popisuje objevování sousedů, na kterých je EIGRP protokol spuštěn, a kteří v tabulce sousedů dosud nemají žádného souseda. Nejprve přidáme síť na směrovač a budeme sledovat vysílání HELLO paketů a s tím spojené hledání sousedů. Po přidání stejné sítě na sousedním směrovači budeme sledovat komunikaci mezi oběma směrovači a změny, které se na nich budou dít. Budou zde popsány struktury HELLO a UPDATE paketů a jejich obsah. Uvidíme, jak si směrovače potvrzují přijetí paketů a co se děje, pokud nebyl paket potvrzen.

Cíle

- Reakce na přidání sítě
- Struktura EIGRP paketu a jeho obsah
- Reakce směrovače na objeveného souseda
- Změny na směrovačích



Obrázek 4.1: Schéma zapojení směrovačů a počítačů s programem Wireshark

- Komunikace mezi směrovači

Postup

1. Konfigurace EIGRP na R25 (síť 172.16.0.0)
2. Konfigurace EIGRP na R26 (síť 172.16.0.0)

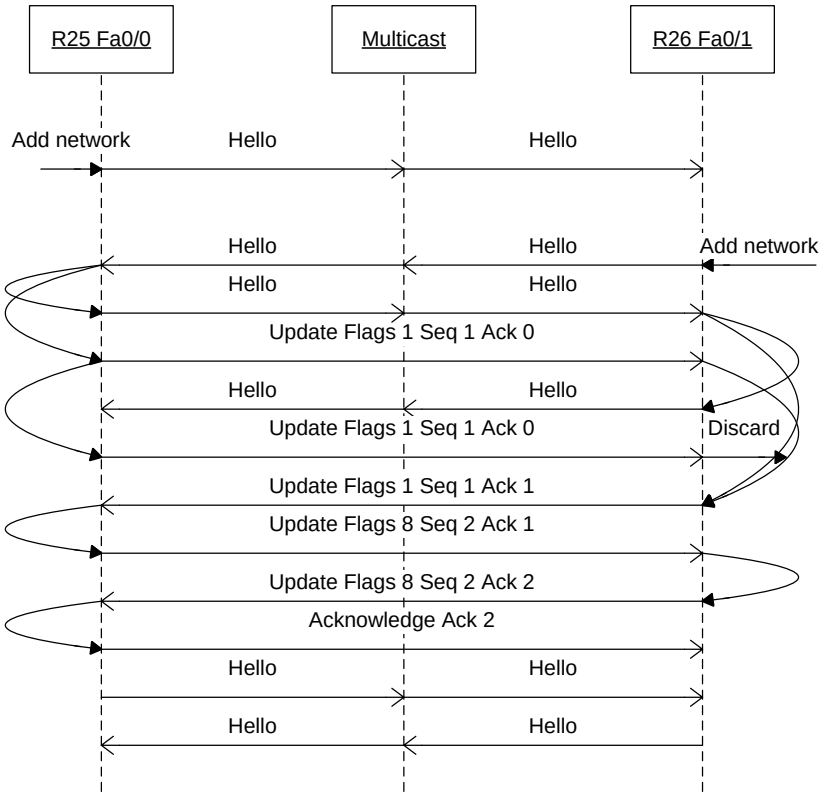
Průběh

- Po zadání příkazu `router eigrp 100` se na síti neprojeví žádná změna. Pouze až po přidání sítě (`network 172.16.0.0`), se do tabulky topologie uloží informace o přístupu do sítě 172.16.00 přes rozhraní Fa0/0, a jeho parametry:
    1. Zjistí, že síť 172.16.0.0/16 není v tabulce topologie a přidá ji s maximální metrikou.
    2. Spočítá feasible distance a reported distance podle přijatých parametrů. Protože jde o vlastní síť je reported distance 0. Feasible distance se spočítá z parametrů rozhraní, na kterém je síť připojena. Tato situace je podrobněji popsána ve scénáři Přidání sítě sousedovi (4.1.2).
    3. Najde feasible successor (ten má maximální metriku).
    4. Protože je nová metrika menší než nalezená, nahradí feasible successor a přidá se do směrovací tabulky.
    5. Změnila se topology table, takže je třeba poslat UPDATE všem sousedům, ale protože žádní nebyli dosud objeveni (refcount je 0), nepošle žádné.
- Dále lze pozorovat rozesílání HELLO paketů s id 88 na multicastovou adresu 224.0.0.10. Pakety jsou rozesílány s intervalem 5s. Struktura paketu odpovídá popisu v kapitole Chování EIGRP, zaměříme se proto na jeho hlavičku:

Parametr	Hodnota	Poznámka
Version	2	
Opcode	5	jde o HELLO paket
Checksum	0xee68	kontrolní součet, který se počítá stějně jako u UDP
Flags	0x00000000	příznaky
Sequence	0	nečeká potvrzení paketu
Acknowledge	0	nepotvrzuje žádný paket
Autonomous System	100	Číslo autonomního systému ve kterém se informace šíří

a obsah:

Parametr	Hodnota	Poznámka
Type	0x0001	Následující obsah jsou Eigrp parametry
Size	12B	Délka obsahy je 12 bytů
K1 - K5	10100	parametry pro výpočet metriky
Reserved		Rezervovaný byte
Hold time	15	čas v sekundách, po jehož vypršení je soused označen za nedostupného



Obrázek 4.2: Komunikace v rámci objevování sousedů

Dále informace o softwaru, které nejsou z hlediska simulace zajímavé.

Protože směrovač nemá žádné sousedy, je tabulka sousedů prázdná. V tabulce topologie je už ale přidaná síť s nulovou metrikou a přístupem přes vlastní rozhraní.

- Po nastavení EIGRP na sousedícím směrovači R26 a přidání společné sítě, lze vysledovat komunikaci, kterou popisuje obrázek 4.2.

- Zde je vidět, že R25 pošle inicializační UPDATE paket 2x, protože nedostal odpověď v požadované době. R26 však paket zachytil a na druhý paket nereagoval, protože paket s tímto sekvenčním číslem již zpracoval (na obrázku znázorněno šipkou Discard).

- Po této operaci se upravila tabulka sousedů na obou směrovačích. Zde je zobrazen jeden záznam z tabulky sousedů na směrovači R25:

Parametr	Hodnota	Poznámka
H	0	byl objeven první soused
Address	172.16.0.2	adresa souseda
Interface	Fa0/0	soused je připojený k rozhraní Fa0/0
HoldTime	13	pokud do 13s nepřijde od souseda HELLO paket, bude tento vymazán (Restartuje se s každým přijatým HELLO paketem od souseda)
SRTT	14	průměrný čas mezi odesláním a přijetím paketu v ms
RTO	3000	v případě selhání multicastu, čeká 3000ms na potvrzení
Q	0	žádné QUEUE pakety ve frontě
Seq	2	sekvenční číslo posledního přijatého paketu (R26 má Seq 3)

Z průběhu lze vysledovat, že k objevování sousedů slouží HELLO pakety. Po výměně HELLO paketů, respektive po obdržení HELLO paketu si směrovače přidají souseda do tabulky sousedů, odešlou mu HELLO paket a začnou mu posílat UPDATE pakety s informacemi o své topologii. Jako první pošlou init UPDATE s Flags = 1. Prázdný UPDATE s Flags 8 je nejspíše uzavírací UPDATE a říká směrovači, že už žádná změna nepřijde. Přijetí paketů se potvrzuje pomocí Acknowledge, který má hodnotu Sequence potvrzovaného paketu. Aby se zbytečně nezatěžovala linka potvrzovacími pakety, je potvrzení posíláno s dalšími informacemi v UPDATE paketu (Ack uloží do fronty a až když nemá update k odeslání pošle HELLO paket s patřičným Acknowledge). Pokud není žádný paket k odeslání, je potvrzení posláno HELLO paketem s potvrzovaným Acknowledge. Tímto způsobem zajišťuje EIGRP spolehlivý přenos dat podobně jako TCP. Důkazem toho je dvojí odeslání inicializačního UPDATE paketu směrovačem R25. Po vypršení RTO je paket znovu odeslán a RTO zvýšeno. Dále byl zjištěn a popsán průběh přidání sítě do EIGRP.

4.1.2 Přidání sítě sousedovi

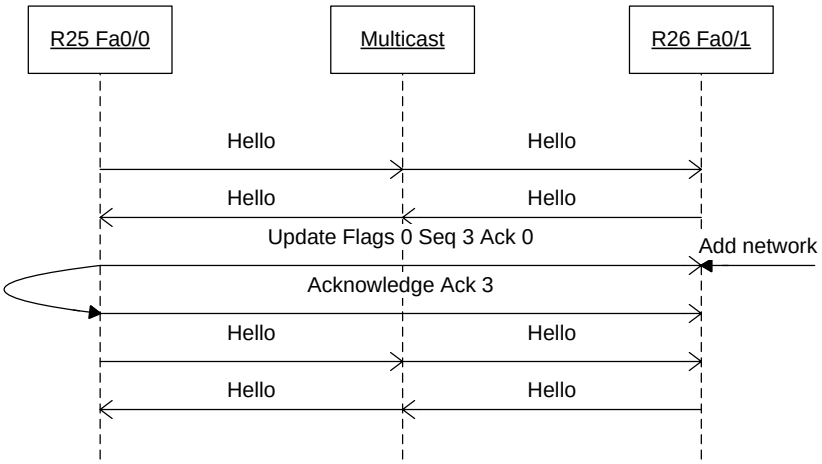
Přidání sítě na sousedovi je doprovázeno šířením informací o této síti. Scénář proto popisuje jednoduchou komunikaci mezi směrovači a struktutu UPDATE paketu s informacemi o síti. Uvidíme zde strukturu této sítě a vysvětlíme si co jednotlivé parametry představují. Dále zde bude popsán výpočet metriky pro přijatou síť a jak je síť přidána do topologické tabulky a směrovací tabulky. Také lze sledovat, jakým způsobem se potvrzují multicast pakety.

Cíle

- Struktura UPDATE paketu se sítí k šíření
- Reakce na UPDATE paket od souseda

Postup

1. Konfigurace EIGRP na R26 (síť 10.0.0.0)



Obrázek 4.3: Přidání sítě sousedovi

Průběh

- Přidání sítě do EIGRP na směrovači R25 probíhá podobně jako v prvním scénáři, protože však nyní je refcount 1 (už máme souseda), pošle na všechna rozhraní, na kterých je připojen alespoň jeden soused UPDATE paket (na multicast) a čeká na potvrzení. S každým přijatým potvrzením sníží refcount o 1 a až když roven 0, odblokuje multicast na rozhraní.
- Po přidání sítě lze pozorovat jednoduchou komunikaci, kterou popisuje obrázek 4.3.
- Směrovač R26 pošle na multicast informace o nové síti. Sekvenční číslo je 3, protože poslední odeslaný paket měl sekvenční číslo 2 a HELLO pakety se nezapočítávají (nepoužívají seq).
- UPDATE paket posílá oznámení o jedné vnitřní síti, které jsou následující:

Parametr	Hodnota	Poznámka
Next Hop	0.0.0.0	pakety posílat na adresu souseda
Delay	2560	podle rozhraní, tabulková hodnota, jde o součet od souseda do sítě
Bandwith	25600	šířka pásma
MTU	1500	maximální velikost IP paketu je 1500B
Hop Count	0	síť je přímo připojena na souseda
Reliability	255	maximální spolehlivost
Load	1	nízké zatížení
Rezervovaný Byte		
Prefix Length	8	maska sítě je 8
Destination	10.0.0.0	adresa sítě (poslala se jen 10, zbytek jsou po konjunkci s maskou 0, takže je není třeba posílat)

- Výše zmíněná data bere R26 ze své topologické tabulky, kam je uloží po přidání sítě 10.0.0.0 do EIGRP.
- Směrovač R25 zařadí potvrzení o přijetí UPDATE paketu do fronty a začne jej zpracovávat:
    1. Zjistí, že síť 10.0.0.0/8 není v topologické tabulce a přidá ji s maximálními metrikami.
    2. Spočítá feasible distance a reported distance podle K parametrů, přičemž parametry pro feasible distance získá tak, že k parametrům z update paketu přidá informace z rozhraní, na kterém je soused připojen. Tyto informace jsou taktéž v tabulce topologie v síti, ve které se soused nachází, ale hledání je náročná operace, proto bych si dovolil odhadnout, že informace získává z rozhraní. Delay sčítá, Bandwith a Reliability použije nejmenší a Load největší. na získáné parametry aplikuje vzorec z kapitoly Výpočet metriky (3.2.1). Z ladícího výpisu `DUAL: rcvupdate: 10.0.0.0/8 via 172.16.0.2 metric 30720/28160` je patrné, že nenásobí vzorec číslem 256. Proč v cisco dokumentu vzorec tímto číslem násobí, se nepodařilo zjistit.
    3. Zjistí, že metrika je menší než nalezená a označí souseda za successor pro danou síť a přidá ji do směrovací tabulky.
    4. Pošle potvrzení o přijetí.
    5. Opět rozešle UPDATE na všechny rozhraní, na kterých má připojeny sousedy, ovšem vynechá rozhraní Fa0/0 odkud UPDATE přišel (split horizon) a sám si tento paket potvrdí, aby snížil refcount, a odblokoval multicast.

Byl zjištěn tvar UPDATE paketů a jejich obsah, upřesněno přidávání sítě do EIGRP a to jak ručně (přímo na směrovači), tak přijatým UPDATE paketem. Ze sledování vyplývá, že celý proces nad EIGRP tabulkami řídí DUAL, kterému jsou pouze předávány vstupní informace. Dále lze vysledovat zamezení šíření aktualizací sousedovi, který dal k aktualizaci podnět. Tato technika se nazývá split horizon.

4.1.3 Zjišťvání sousedů s jedním sousedem

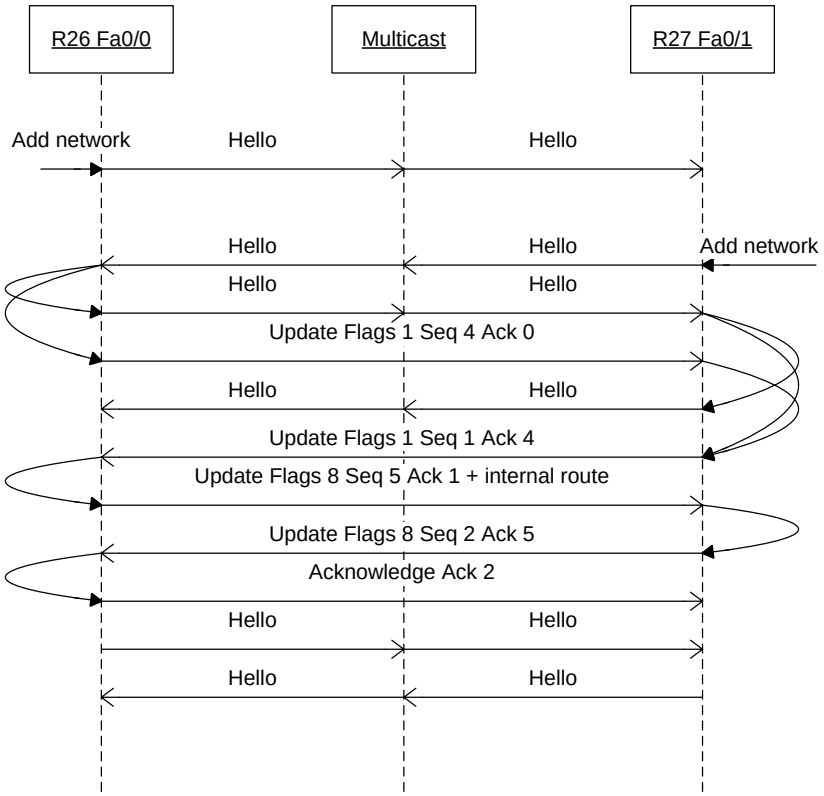
Scénář ukazuje, jak vypadá komunikace mezi sousedy, kdy v rámci procesu objevování sousedů pošle místo prázdných UDPATE paketů informace o ostatních sítích. Bude zde vysvětleno, proč v prvním případě zjišťování sousedů byly UPDATE pakety prázdné. Dále zde bude vysvětleno odkud bere směrovač sekvenční čísla odesílaných paketů.

Cíl

- Zjistit odlišnosti mezi zjišťováním sousedů bez a se sousedem

Postup

1. Konfigurace EIGRP na R27 (síť 10.0.0.0)



Obrázek 4.4: Komunikace mezi sousedy, kdy jeden již souseda (a jeho sítě) zná

Průběh

- Do EIGRP na R27 se přidá síť 10.0.0.0 (postup je stejný jako u prvního příkladu)
- Komunikaci popisuje obrázek 4.4

Z komunikace lze vypozorovat dvě změny. Prví je ta, že první UPDATE paket od směrovače r26 nemá seq 1, ale 4, z čehož vyplývá, že toto číslo je drženo pro celé EIGRP na daném směrovači (3 pakety už poslal na R25) a ne ke každému sousedovi zvlášť. Druhá změna spočívá v tom, že místo prázdného paketu s flags 8 pošle R26 paket s cestou do sítě 172.16.0.0, z čehož usuzuji, že

1. tento paket musí být odeslán
2. nešíří informace o síti, na které komunikace probíhá

4.1.4 Odstranění sítě bez souseda

Scénář popisuje nejjednodušší odstranění šíření informace o síti ze směrovače, který už nemá žádného souseda, kterému by změnu posílal. Protože nemá žádného souseda, nerozešle ani QUERY pakety s dotazem na odstraňovanou síť. Nakonec jen smaže EIGRP rozhraní a ukončí odesílání HELLO paketů.

Ačkoliv skutečný postup byl trochu jiný, dovolím si v případě odstraňování sítí k šíření začít od konce, tj. odebráním poslední sítě v schématu, abychom mohli pozorovat a popisovat dění od jednodušších akcí po ty náročnější.

Cíl

- Zjistit změny na směrovači po odebrání sítě k šíření.

Postup

1. Pomocí příkazu `no network 192.168.1.0` odebereme poslední síť směrovači R25

Průběh

- DUAL obdrží aktualizaci od „Connected“ (tj. vlastní rozhraní) o smazané síti s feasible a reported distance s maximální hodnotou.
- DUAL hledá feasible successor pro tuto síť - nenalezen, minimální vzdálenost je nastavena na maximum
- Protože nemá žádné sousedy, nepřechází do aktivního stavu.
- DUAL přejde do stavu ifdelete
- Smaže EIGRP interface.
- DUAL ukončí stav ifdelete.

Sítě přímo připojené nemají v tabulce topologie odkaz na souseda, ale jsou označeny jako Connected a jejich feasible distance je metrika rozhraní, na kterém je síť nastavena. Mazání probíhá tak, že se DUALu pošle aktualizace o této síti, kde jako odesílatel je uvedeno Connected a metriky jsou maximální. Pak DUAL zkusí najít jiný feasible successor a když neuspěje, smaže rozhraní a ukončí se.

4.1.5 Odstranění sítě se sousedem

Zde budeme pozorovat, jakým způsobem dá směrovač vědět svým sousedům, že končí s šířením své sítě. Uvidíme strukturu Goodbye paketu, jeho obsah a jak na tento paket soused reaguje. Zde již lze pozorovat snaha rozeslat QUERY paket, která je ukončena odebráním posledního souseda a jeho záznamů. Když není soused, který by paket přijal, nepošle se.

Cíle

- Zjistit, jak dá směrovač vědět sousedovi, že končí s šířením EIGRP

Postup

1. Odebereme síť 192.168.1.0 na směrovači R27

Průběh

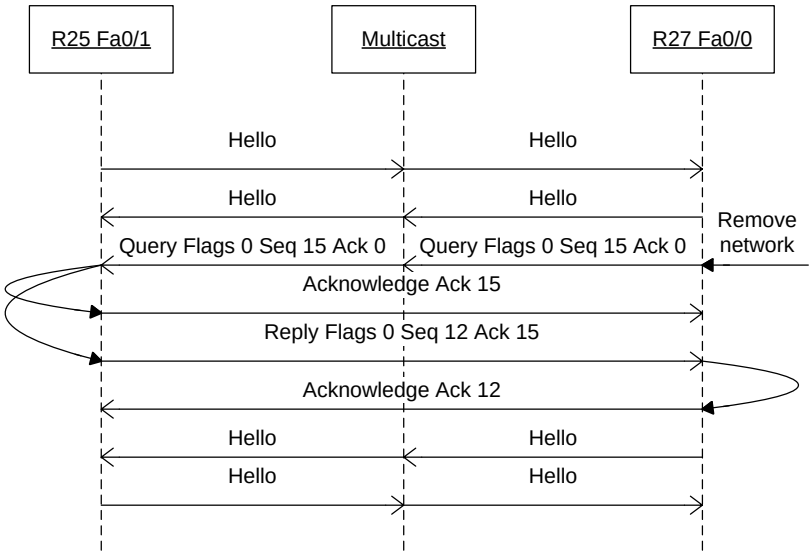
Průběh na R27

- DUAL opět obdrží aktualizaci od Connected, ale tentokrát má 1 souseda (peer1) a tak přejde do aktivního stavu
- DUAL vytrvoří tabulku stavu odpovědí s 1 řádkem (má 1 souseda)
- Pošle na multicast Hello paket s K parametry 255, což je Goodbye paket, kterým sousedům říká, že končí.
- DUAL - změna souseda, rozhraní Fa0/0 je vypnuto
- Tabulka sousedů vypne všechny sousedy za rozhraním Fa0/0
- Dual přejde do stavu vypínání linky (linkdown)
- Vypnutí souseda musí být předáno tabulce odpovědí, aby mohla vymazat řádky s nedostupnými sousedy
- tabulka odpovědí je prázdná, uvolní ji
- Vyhledá feasible successor pro síť 192.168.1.0 - najde s maximální metrikou
- Smaže síť z tabulky topologie, soused je vypnutý, poslední soused byl smazán, odstraní refcount (refcount = 0)
- DUAL ukončí stav linkdown
- DUAL přejde do stavu ifdelete
- Smaže EIGRP interface.
- DUAL ukončí stav ifdelete.

Průběh na R25

- Obdrží Goodbye paket
- Vypne souseda, důvod: obdržení Goodbye paketu
- DUAL přejde do stavy linkdown (s parametrem 192.168.1.0)
- Zjistí, že byl smazán poslední soused (kdyby byly nějaké zbývající záznamy k tomuto sousedovi, patrně je vymaže)
- DUAL konec stavu linkdown

Směrovač, který má alespoň jednoho souseda a končí s šířením sítě, pošle na multicast HELLO paket s maximálními K parametry, který sousedící směrovač označí za Goodbye paket a odesílatele označí za vypnutého. Toto se děje pouze v rámci EIGRP, se skutečnými rozhraními to nesouvisí. Protože DUAL odebírá síť a má souseda, chce odesílat QUERY pakety. Před odesláním, ale souseda vypne, což má za následek smazání dotazu na síť. Tabulka dotazů je prázdná a tak se doptávání na síť ukončí. Nakonec se odebere EIGRP rozhraní, na které je odebíraná síť připojena.



Obrázek 4.5: Hledání cesty do cíle

4.1.6 Odstranění sítě se sousedem a neexistující cestou

Na tomto scénáři budeme pozorovat poptávání se po síti, na kterou směrovač ztratil cestu. Uvidíme zde, jak síť přechází do aktivního stavu a jak odesílá QUERY pakety. Ukážeme si strukturu tohoto paketu a jeho obsah jako rozdíl oproti UPDATE paketu. Dále budeme moci pozorovat reakci sousedního směrovače na QUERY paket a jeho odpověď paketem REPLY. Dozvíme se zde, že QUERY paket je potvrzován okamžitě a ne až v REPLY paketu jako tomu bylo v procesu objevování sousedů. Opět bude ukázána struktura a obsah REPLY paketu.

Cíle

- Zjistit, jak se v rámci EIGRP šíří informace o odebrané síti.
- Parametry QUERY paketu
- Parametry REPLY paketu

Postup

1. Na směrovači R27 odebereme síť 10.0.0.0

Průběh

- Odstranění sítě proběhne stejně jako v předchozím případě. Důležitý je přechod do aktivního stavu, kdy se snaží najít jinou cestu do odstraňované sítě.
- Na neodebraná rozhraní se zašle na multicast QUERY paket s dotazem na odstraněnou síť viz. obrázek 4.5.

- Paket má podobný tvar jako UPDATE paket s následujícími změnami:

Parametr	Hodnota	Poznámka
Opcode	3	jde o QUERY paket
Delay	maximální hodnota	Označuje, že cíl je nedostupný

  Ostatní informace pro výpočet metriky jsou nulové

- R25 pošle potvrzení o přijetí.
- R25 vyhledá feasible successor (vynechá tázajícího se souseda).
- R25 nenajde jinou cestu, odpoví REPLY paketem s informací, že je cíl nedostupný:

Parametr	Hodnota	Poznámka
Opcode	4	jde o REPLY paket
Delay	maximální hodnota	Označuje, že cíl je nedostupný

  Parametry pro výpočet metriky jsou parametry rozhraní.

- Po potvrzení přijetí odpovědi vymaže R25 z topologické tabulky (a pravděpodobně i ze směrovací tabulky)

Po odstranění sítě (ve skutečnosti přímé cesty do sítě) přejde směrovač do aktivního stavu, kdy se snaží vyhledat jinou cestu. Pošle dotaz na multicast, na kterém ostatní směrovače zkusí najít jinou cestu a v REPLY pakety vrátí cestu s novou metrikou, nebo s informací, že je cíl nedostupný. Protože vyhledávání může být náročný proces, potvrzují se jednotlivé pakety okamžitě.

4.1.7 Odstranění sítě se sousedem a existující cestou

Protože v předchozím případě jsme nenašli cestu do cíle, bylo by dobré uvést si, jak vypadá kladná odpověď na dotaz. Uvidíme zde také, jak se pomocí QUERY paketů šíří informace o síti, a jaké změny na směrovači proběhnou, pokud přijetím tohoto paketu zjistí, že cesta, kterou používal, je nedostupná.

Cíl

- Jak vypadá situace, kdy soused zná cestu do sítě.

Postup

1. Odebereme síť 172.16.0.0 na R25.

Průběh

- Po odebrání sítě a souseda, je třeba najít jiné cesty do sítí, které byly odebrány, nebo připojeny za odebraného souseda.
- Pro síť 10.0.0.0 nalezen feasible successor přes 192.168.1.2, odstraní starý successor a nahradí jej nově nalezeným. Taky upraví směrovací tabulku.

- O změně chce poslat UPDATE paket oběma sousedům, ale protože jeden byl zvolen za feasible successora a druhý je odstraněn, nepošle žádný.
- Pro síť 172.16.0.0 nenalezne feasible successor, a tak pošle QUERY paket.
- R27 feasible successor najde a pošle QUERY s metrikou.
- Protože R27 používalo ztracenou cestu jako přístup do sítě 172.16.0.0, musí upravit směrovací tabulku a polstat UPDATE paket o změně. Paket posílá na multicast na obě rozhraní.
- R26 na něj nereaguje, protože do cíle má vlastní cestu, jen pošle potvrzení o přijetí.
- Jakmile obdrží R25 REPLY paket, vyčistí tabulku odpovědí a hledá feasible successor.
- Protože nechce šířit informace o síti, má nalezený feasible successor maximální metriky a síť 172.16.0.0 je odstraněna z tabulky topologie.
- Dále už jen potvrdí přijaté pakety.

Odstranění sítě probíhá podobně jako v předchozím případě, pouze R25 zkusí najít jinou cestu do odebrané sítě. Směrovače R26 a R27 feasible successor do sítí naleznou ve svých topologických tabulkách a tak QUERY pakety posílat nemusí. Pouze pokud nastala změna, jako v případě R27, odešle směrovač na multicast UPDATE paket.

4.1.8 Přidání souseda se známou cestou

Zde uvidíme jak reaguje směrovač na obdržení UPDATE paketu se sítí, do které patří jedno z jeho rozhraní, ale o které nechce šířit informace. Po přidání sítě k šíření, budeme moci pozorovat odstranění cest z tabulky topologie, které vedou do stejné sítě jako připojené rozhraní. Dále lze pozorovat, že po výměně informací o sítích se sousedy se aktivuje proces poison reverse, který zabrání nekonečné smyčce šíření informací.

Cíle

- Zjistit, jakým způsobem reaguje směrovač na přidání nové cesty do místa, kam už cestu zná.

Postup

1. Konfigurace EIGRP na R27 (síť 192.168.1.0)
2. Konfigurace EIGRP na R25 (síť 192.168.1.0)

Průběh

- Přidání sítě na R27 proběhne jako obvykle s tím rozdílem, že po přijetí UPDATE paketu na R26 (a šíření informací o síti k R25) odešle R26 na multicast rozhraní Fa0/0 UPDATE paket s informací, že tato síť je přes R26 nedostupná. Tento jev se nazývá poison reverse. Že se tento jev neobjevil v případě přidání sítě 10.0.0.0 na R26 si vysvětluji tím, že pokud nemá směrovač sousedy a tudíž nehrozí nebezpečí nekonečné smyčky, poison reverse neprovádí.

- Směrovač pošle QUERY o přidávané síti. Toto si vysvětluji tím, že po přidání cesty, která je přímo připojená, dojde k odstranění záznamů z tabulky topologie, kde cíl je stejný jako přidávaná cesta. A jako všecha odstranění, provází i tuto situaci poptávání QUERY pakety po jiných cestách. Toto se děje pravděpodobně kvůli šíření lepší metriky.
- Ustanoví sousedství s R27.
- Pošlou si UPDATE pakety a na multicast UPDATE pakety s nedostypnými cestami (poison reverse)
- R25 pošle na multicast rozhraní Fa0/1 UPDATE paket s informací o síti 192.168.1.0 a nedostupnosti sítě 10.0.0.0 (nyní má poison reverse smysl)
- R26 potvrdí přijetí a pošle informaci o nedostupnosti sítě 192.168.0.0
- Výměna informací o nedostupnosti je stejná pro všechny směrovače.

Z tohoto scénáře vycházejí dva důležité poznatky. První je ten, že při přidání přímo připojené sítě se pravděpodobně odstraní záznamy o této síti z tabulky topologie, a s tím je spojený důsledek počátku dotazování se směrovače na síť. Druhý důležitý poznatek je, že pokud má směrovač více než jednoho souseda, aplikuje techniku poison reverse, která dělá to, že pokud směrovač přijme informace o síti a označí souseda za successor, rozešle informace o nedostupnosti sítě přes sebe na multicast rozhraní, na kterém je soused připojen. Tímto se zabrání smyčce v posílání informací o síti, protože pro danou cestu existuje v síti jen jeden směrovač.

4.2 Návrh implementace protokolu EIGRP

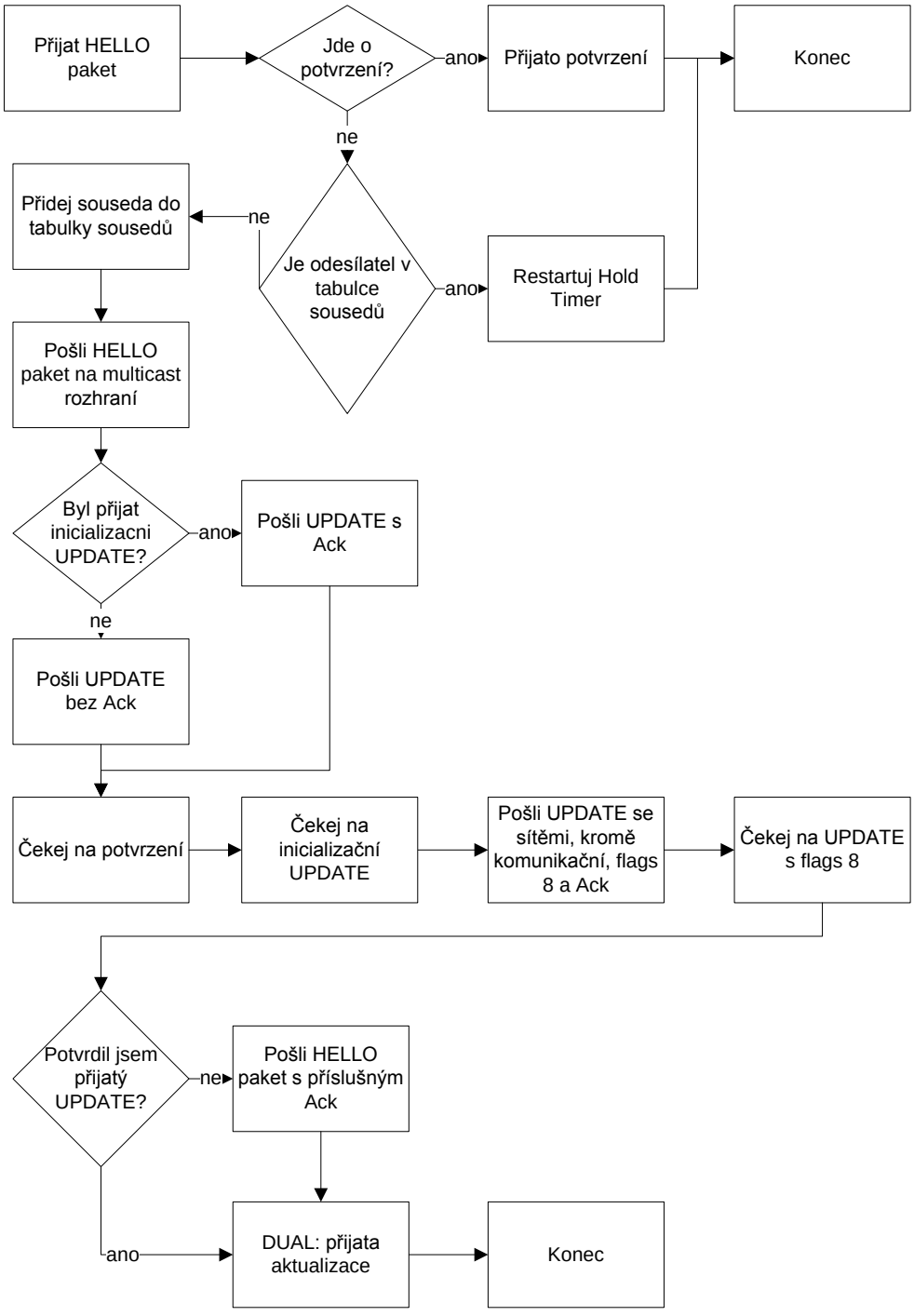
Na základě analýzy komunikace, kterou jsme měli možnost sledovat, a ladících výpisů, jsem se pokusil definovat procesy, které ve směrovači probíhají. Jsou zde pomocí diagramů popsány reakce na vybrané události. Obdržení paketů je popsáno v klidovém stavu, pokud není určeno jinak.

4.2.1 Slovník pojmů

Tabulka 4.2.1 popisuje zkratky uvedené v diagramech, jejich anglický překlad a stručný popis, co která zkratka znamená.

4.2.2 Přijat HELLO paket

Proces přijetí HELLO paketu znázorňuje diagram na obrázku 4.6. Nejprve ověří, jestli se jedná o potrzovací paket, který má nenulové Acknowledge, a spustí proces `Přijato potvrzení`. Dále ověří, jestli je odesílatel v tabulce sousedů, a pokud ano, restartuje jeho Hold Down timer. V opačném případě přidá nového souseda do tabulky sousedů. Aby ho druhý soused ihned zaregistroval, pošle na multicast HELLO paket a přímo sousedovi inicializační UPDATE paket. Po výměně inicializačních UPDATE paketů, si sousedé vymění UPDATE paket se sítěmi z topologické tabulky, kromě sítě, na které spolu komunikují. Všechny pakety musí být potvrzeny.



Obrázek 4.6: Proces Přijato HELLO paket

Výraz	Anglický výraz	Popis
RT	routing table	směrovací tabulka
TT	topology table	tabulka topologie
S	successor	cesta do sítě, kterou směrovač používá (je ve směrovací tabulce)
FS	feasible successor	záložní cesta do sítě
FD	feasible distance	metrika nejlepší cesty do sítě
ReT	reply table	tabulka odpovědí na odeslaný QUERY dotaz
AcT	acknowledge table	tabulka odeslaných paketů, čekajících na potvrzení
Connected		připojené rozhraní
cesta		struktura zahrnující cíl cesty, souseda, přes kterého je dosažitelná a metriku
maximální metrika		nekonečná metrika, reprezentovaná maximálním možným číslem

Tabulka 4.1: Slovník pojmů

4.2.3 Přijat QUERY paket

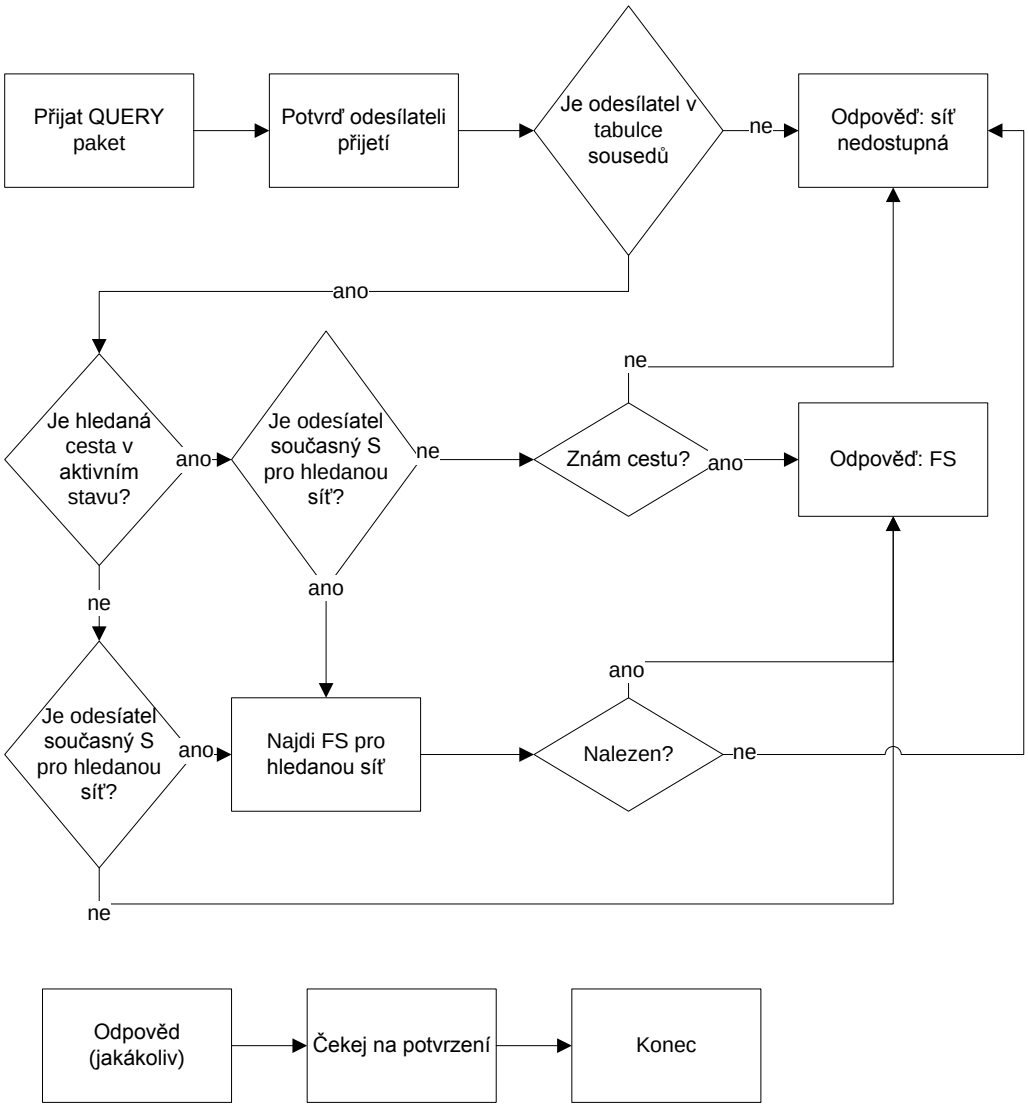
Proces zpracování přijatého QUERY paketu znázorňuje diagram na obrázku 4.7. Proces potvrdí přijetí paketu odesláním HELLO paketu s Acknowledge rovnou sekvenčního čísla QUERY paketu. Zjistí, jestli je odesílatel v tabulce sousedů, pokud ne, odpoví REPLY paketem, že síť je nedostupná. Pokud ano, tak v případě, že je síť je v aktivním stavu a soused je pro tuto síť successor, pokusí se najít jiný feasible successor a výsledek pošle jako odpověď. Pokud není successor, vrátí výsledek podle toho, jestli successora zná. Pokud síť není v aktivním stavu a soused není successor, vrátí mu současný successor, pokud je successor, musí vyhledat jiný a výsledek vrátí. Pro hledání feasible successora spouští proces `Najdi FS` s parametrem adresy dotazované sítě.

4.2.4 Přijato potvrzení

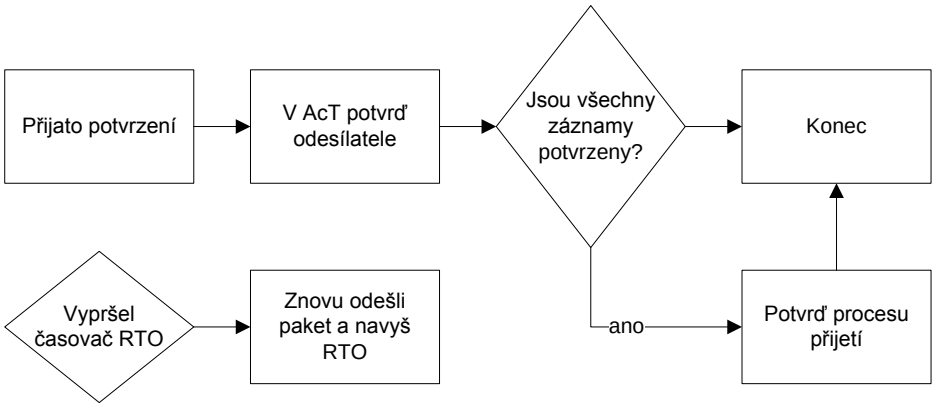
Přijetím jakéhokoliv paketu s nenulovým Acknowledge se spustí proces potvrzení paketu viz. diagram na obrázku 4.8. Ten vyhledá v tabulce odeslaných paketů odeslaný paket podle jeho sekvenčního čísla a potvrdí jeho odeslání. Protože jsou některé pakety posílány na multicast (například QUERY, UPDATE), je třeba s jejich odesláním nastavit počet sousedů, a s každým potvrzením tento počet snížit. Tak docílíme toho, že budou všechny pakety doručeny. S odesláním paketu také spustíme časovač RTO. Pokud vyprší dříve než je paket potvrzen, dojde k opětovnému odeslání paketu a navýšení RTO. K tomuto jevu dochází proto, aby v případě vyššího zatížení linky směrovač zbytečně neposílal každý paket dvakrát. V případě pádu linky, a s tím spojené odstraňování souseda, je jeho záznam vymazán, aby neblokoval čekající procesy.

4.2.5 DUAL: přijata aktualizace

Diagram na obrázku 4.9 popisuje zpracování UPDATE paketu a přidání/odebrání sítě přímo na směrovači, kdy je jako odesílatel označen Connected, což je rozhraní směrovače, které patří do šířené sítě. Jako parametr vyžaduje cestu UP do sítě s metrikou. Inicializace zá-



Obrázek 4.7: Proces Přijat QUERY paket



Obrázek 4.8: Proces Přijato potvrzení

znamu znamená, že se v tabulce topologie vytvoří záznam pro cíl, a jeho feasible distance a reported distance se nastaví na maximální hodnotu. Dále pak zjišťuje, jestli odesílatel aktualizace UP je přímo rozhraní směrovače, a pokud ano, odstraní dosud vedené záznamy o cestě, a začne hledat nový feasible successor. Toto chování bylo vypozorováno z měření. Pokud se cesta z UP v tabulce topologie už nalézá, upraví se její záznam a pro vyhledání nového feasible successora zavolá proces `Najdi FS` s přidaným parametrem přijaté aktualizace UP. Pokud je přijat UPDATE paket, je nutné zaslat potvrzení pomocí HELLO paketu, a procesu DUAL postupně předávat jednotlivé sítě.

4.2.6 Zaslání QUERY paketu

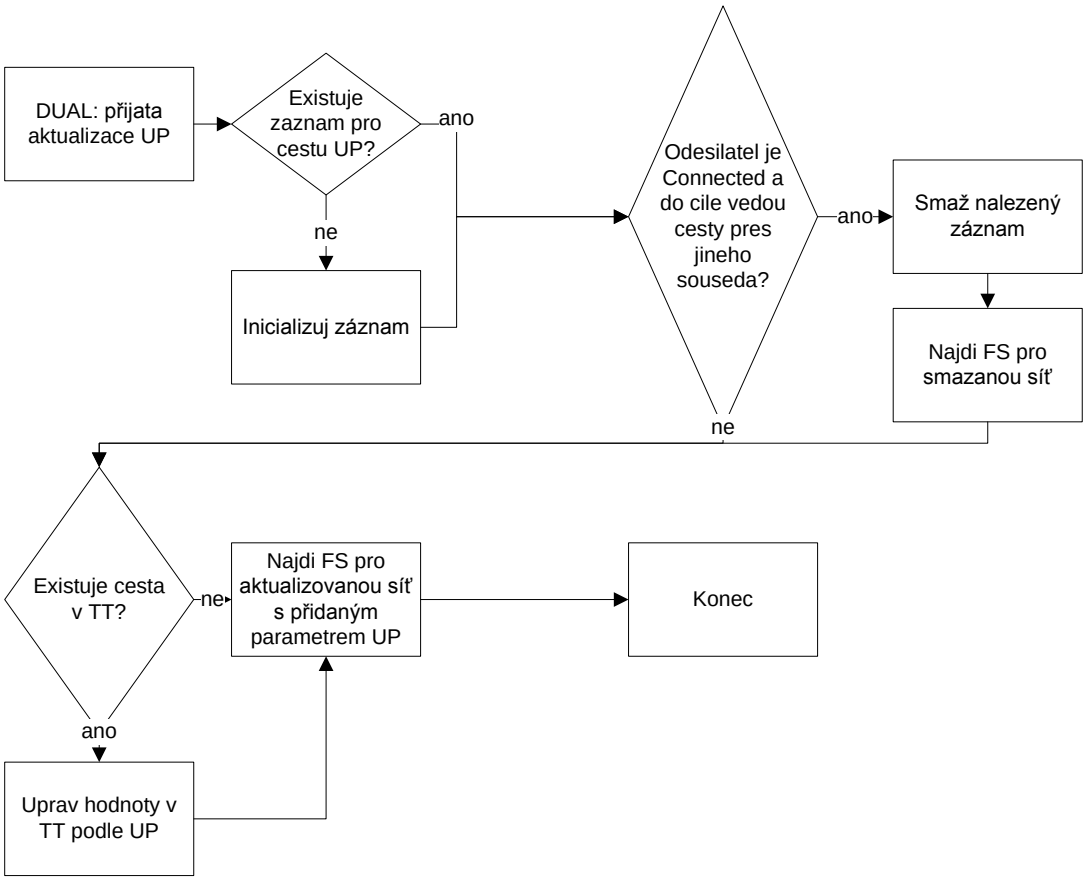
Proces Zaslání QUERY paketu je popsán diagramem na obrázku 4.10 a probíhá tak, že přidá všechny sousedy do tabulky odeslaných paketů, kde budou čekat na potvrzení o přijetí. Dále pak každé rozhraní směrovače, kde je spuštěn EIGRP pošle na multicast QUERY paket s informacemi o síťi, kterou hledá. Pokud je adresa rozhraní shodná s adresou hledané sítě, paket neposílá, a potvrdí přijetí sousedů napojených na rozhraní. Jako parametr přijímá síť NET, na kterou se dotazuje.

4.2.7 Zaslání UPDATE paketu

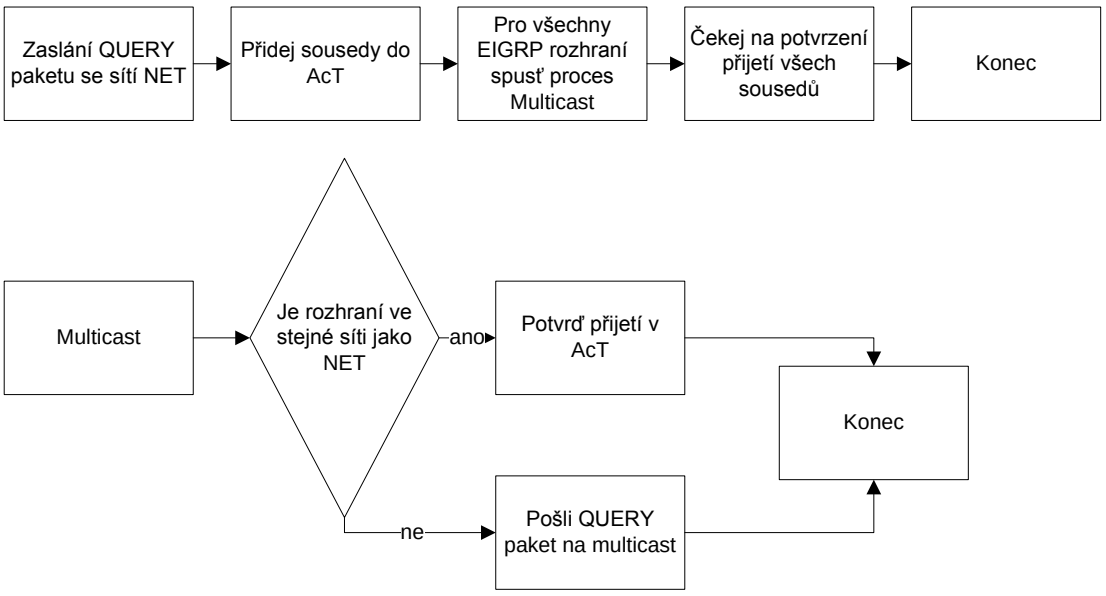
Proces zaslání UPDATE paketu na obrázku 4.11 je podobný jako posílání QUERY paketu. Jako parametr přijímá cestu NET, kterou šíří.

4.2.8 Úprava tabulek

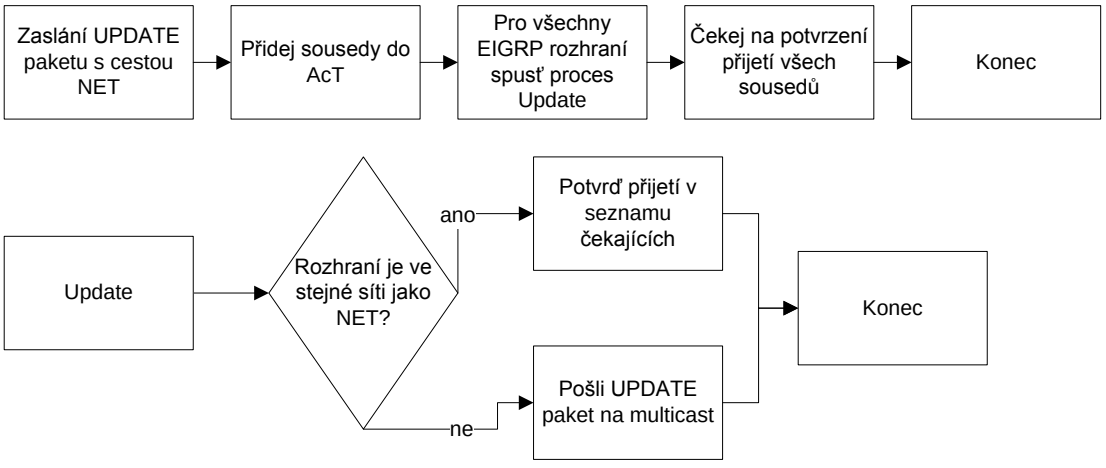
Procesu úprava tabulek viz. diagram na obrázku 4.12 je třeba předat nový záznam, který má být vložen, a starý, který má být vymazán. Proces upravuje tabulku topologie a směrovací tabulku. Odstraní starý záznam, vloží nový, a pokud má směrovač více než 1 souseda, provede poison reverse. Při této operaci na zařízení, na které je připojen soused, od kterého záznam pochází, pošle na multicast UPDATE paket s informací, že tato síť je přes něj nedostupná.



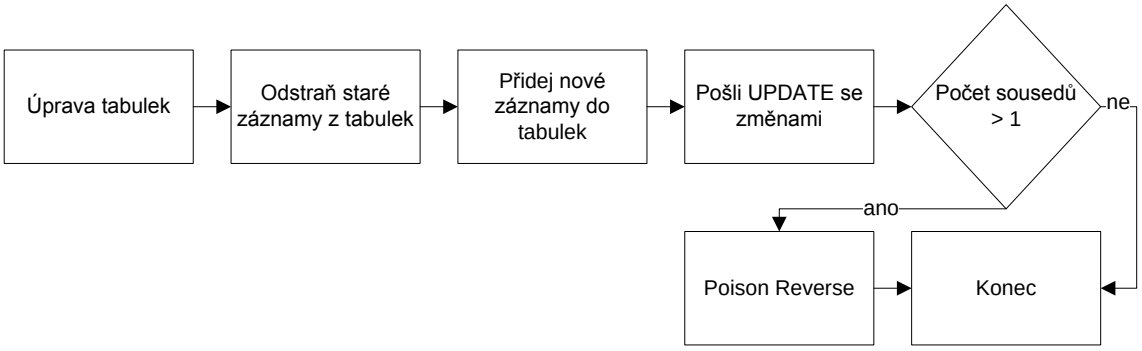
Obrázek 4.9: Proces DUAL: přijata aktualizace



Obrázek 4.10: Proces Zaslání QUERY paketu



Obrázek 4.11: Proces Zaslání UPDATE paketu



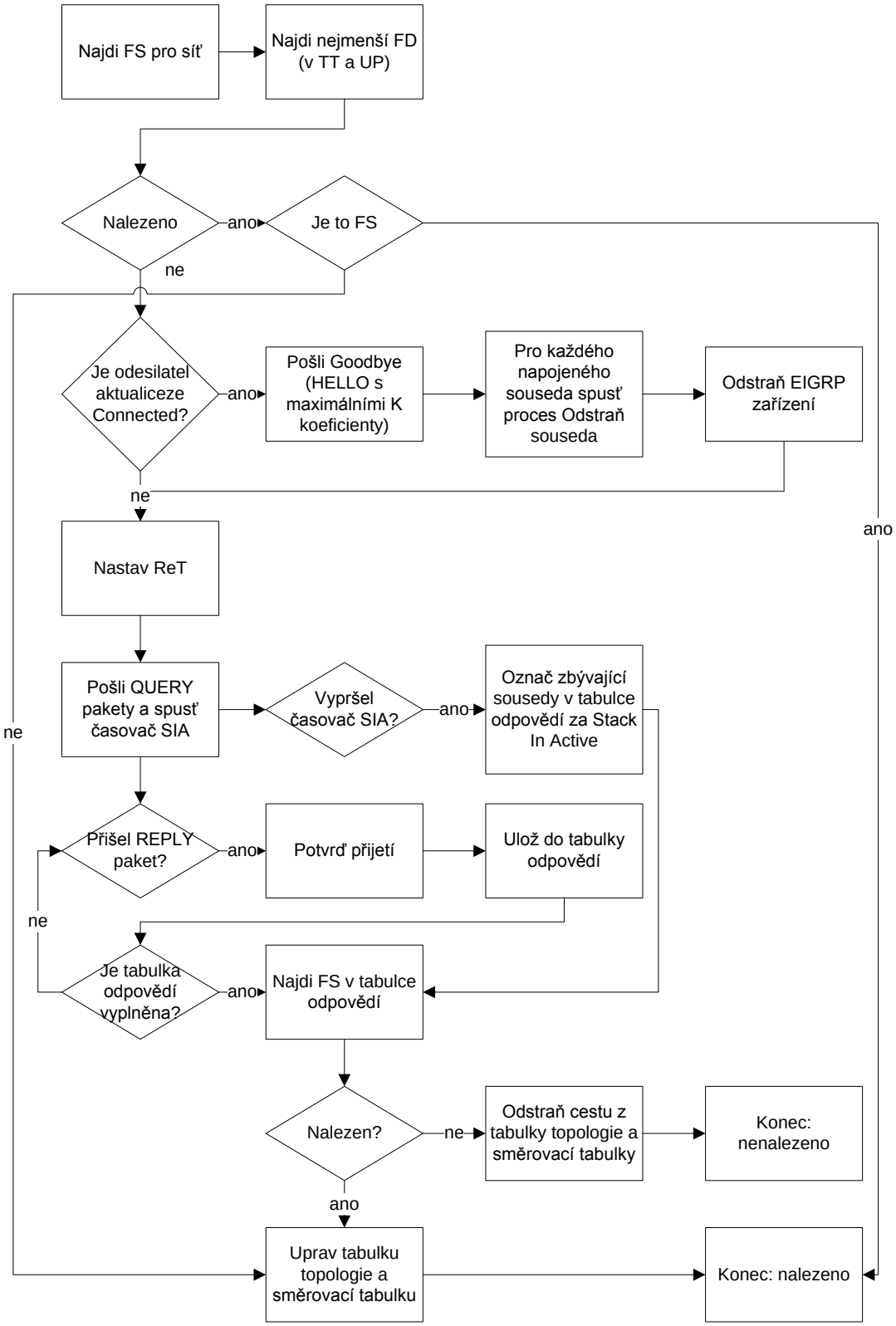
Obrázek 4.12: Proces Úprava tabulek

4.2.9 Najdi FS

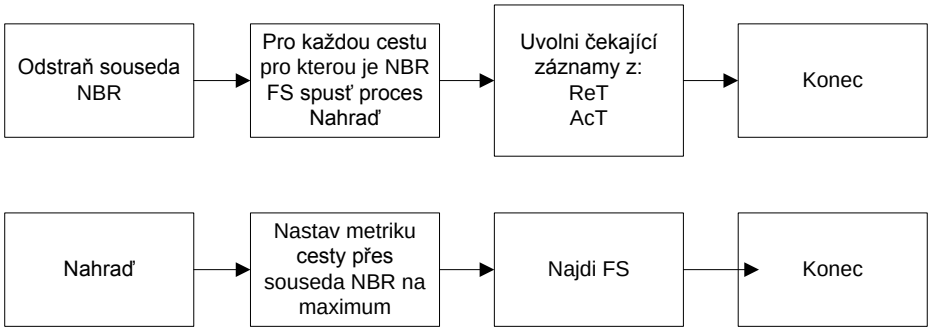
Proces na obrázku 4.13 vyžaduje adresu cíle a nepovinný parametr aktualizace UP, která se přidá k porovnávaným cestám pro vyhledání nejlepší feasible distance. Vyhledá feasible successor pro danou síť tak, že hledá nejlepší feasible distance. Pokud je nalezena (nemá maximální metriku), a jedná se o dosavadní successor, ukončí se s výsledkem cesta nalezena. Pokud není dosavadní successor, zavolá proces `Úprava tabulek` s parametrem nalezené cesty. Když žádný feasible successor nenalezne, ověří, jestli nejde o odstranění sítě. Podívá se na odesílatele aktualizace UP, a když se jedná o přímo připojené rozhraní (Connected), pošle na multicast tohoto rozhranní HELLO paket s hodnotami koeficientů K1 - K5 rovými 255. Jedná se o Goodbye paket. Dále odstraní všechny sousedy připojené přes toto rozhraní, a nakonec odstraní vlastní rozhraní. Odstraněním vlastního rozhraní se myslí pouze EIGRP rozhraní, nikoliv fyzické rozhraní. Protože jsme přišli o cestu do cíle, je nutné se poptat sousedů. Nastaví se tabulka odpovědí, síť přejde do aktivního stavu, a na multicast všech rozhraní pošle QUERY paket s dotazovanou sítí. S touto akcí je spojeno spuštění časovače SIA, který po vypršení označí souseda za Stack In Active a odstraní jej z tabulky sousedů. Po přijetí odpovědi se do tabulky sousedů vrátí. Jakmile je tabulka odpovědí vyplněna, vyhledá v odpovědích feasible successor, a pokud ani zde nenalezne jinou, než maximální metriku, odstraní cestu ze směrovací tabulky a tabulky topologie, a síť se vrátí do pasivního stavu. V případě úspěšného hledání tyto tabulky upraví podle nalezené cesty. Spustí proces `Úprava tabulek` s parametrem nalezené cesty.

4.2.10 Odstraň souseda

Diagram na obrázku 4.14 popisuje proces odebrání jednoho souseda z tabulky sousedů. Jako parametr přijímá odstraňovaného souseda NBR. Aby mohl být soused odstraněn, je nutné projít všechny cesty, kde je soused feasible successor, a pro každou spustit proces `Nahraď` s parametrem této cesty. Proces `Nahraď` nastaví metriku cesty na maximum a spustí proces hledání jiné cesty `Najdi FS`. Jako parametr mu předá adresu cíle cesty. Protože může jiný proces čekat na odpověď od odstraňovaného souseda, je nutné odstranit tyto záznamy z potvrzovací tabulky a tabulky odpovědí.



Obrázek 4.13: Proces Najdi FS



Obrázek 4.14: Proces Odstraň souseda

Kapitola 5

Závěr

Práce se zabývá simulováním sítí, proto je zde popsáno co to taková simulace je, jaké jsou dostupné nástroje pro simulaci a jejich výhody a nevýhody. Podrobněji jsme se seznámili s nástrojem OMNeT++ a jeho rozšířením INET Framework. Je zde popsáno jak takovou simulaci vytvořit, jak řídit její průběh a jak sbírat informace o průběhu k pozdější analýze a zobrazení.

Práce se ale především zabývá analýzou protokolu EIGRP. Protože jde o uzavřený protokol bez veřejné specifikace, bylo nutné zjistit jakým způsobem EIGRP skutečně funguje.

Nejprve jsme sesbírali veřejně dostupné informace o tomto protokolu, abychom je mohli konfrontovat s hodnotami naměřenými v laboratoři. Tyto informace také slouží k pochopení některých jevů a struktur, které jsme pozorovali.

Dále bylo nadefinováno zapojení směrovačů a počítačů, které monitorují komunikaci mezi jednotlivými směrovači. Taktéž byly na směrovačích zapnuty ladící výpisy protokolu EIGRP a jejich ukládání pro pozdější rozbor. Postupně byl EIGRP aktivován na jednotlivých rozhraních směrovačů, a ve stejném pořadí i deaktivován, abychom mohli vysledovat reakce na tyto změny. Záznam komunikace a ladící výpisy jsou uloženy na přiloženém CD. V této části jsou podrobně popsány jednotlivé scénáře, kde je rozebrána komunikace mezi směrovači a změny, které se udály. Tento rozbor pak posloužil ke tvorbě diagramů procesů na směrovači, které by se měly blížit skutečnému chování EIGRP. Každý proces je popsán tak, aby bylo možné jej jednoduše implementovat.

Dalším pokračováním této práce by měla být implementace problému a začlenění do simulačního prostředí. Pro začlenění do prostředí OMNeT++ je nutné vytvořit modul EIGRP a napojit jej na bránu síťové vrstvy v souboru NetworkLayer.ned. To se provede tak, že se vytvoří další dvě brány (vstupní a výstupní) a propojí se s komunikačními bránami modulu. Aby se EIGRP protokoly dostaly do EIGRP modulu je nutné v submodulu IP přidat do parametru `protocolMaping` mapování portu 88 na bránu, kde je EIGRP modul napojen. Dále je potřeba přihlásit všechny směrovače k odebírání EIGRP multicastu 214.0.0.10 a to tak, že se do konfiguračního souboru směrovače (.irt) přidá do parametru `Groups` adresu multicastu. Aby mohl EIGRP modul pracovat, musí implemementovat rozhraní cModule. Při inicializaci musí počkat, až budou naplněny tabulky rozhraní a směrovací tabulka. Musí implementovat funkci, která vrací počet stavů, kterými musí EIGRP projít a v metodě initialize počkat na stav 4. Pak může načíst informace z xml a provést konkrétní kroky, jako inizializace EIGRP rozhraní, zasílání HELLO paketů, atd. Nelze však vyloučit, že CISCO řeší některé procesy jiným způsobem. Proto je nezbytné provést validaci implementace.

Literatura

[1] CISCO: Enhanced Interior Gateway Routing Protocol. 2005, document ID: 16406.

[2] WWW stránky: Data Network Resource. http://www.rhyshaden.com/eigrp.htm.

[3] WWW stránky: NS-2. http://nsnam.isi.edu/nsnam/index.php/Main_Page.

[4] WWW stránky: Omnet++. http://www.omnetpp.org.